DOCUMENT RESUME

ED 127 992 IR 003 961

TITLE	Guidelines for Documentation of Computer Programs and Automated Data Systems. (Category: Software; Subcategory: Documentation).
INSTITUTION	National Bureau of Standards (DOC), Washington, D.C. Inst. for Computer Sciences and Technology.
REPORT NO	DOC-NBS-FIPS-38
PUB DATE	15 Feb 76
NOTE	53p.
AVAILABLE FROM	Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402 (SD Catalog No. C13.52:38; Stock No. 003-003-01580-6; $1.35)
JOURNAL CIT	Federal Information Processing Standards Publication; Feb 1976
EDRS PRICE	MF-$0.83 HC-$3.50 Plus Postage.
DESCRIPTORS	*Computer Programs; *Documentation; Electronic Data Processing; Federal Government; *Guidelines; Information Processing; Standards

ABSTRACT

These guidelines provide a basis for determining the content and extent of documentation for computer programs and automated data systems. Content descriptions of ten document types plus examples of how management can determine when to use the various types are included. The documents described are (1) functional requirements documents, (2) data requirements documents, (3) system/subsystem specifications, (4) program specifications, (5) data base specifications, (6) users manuals, (7) operations manuals, (8) program maintenance manuals, (9) test plans and (10) test analysis reports. (KB)

****
* Documents acquired by ERIC include many informal unpublished *
* materials not available from other sources. ERIC makes every effort *
* to obtain the best copy available. Nevertheless, items of marginal *
* reproducibility are often encountered and this affects the quality *
* of the microfiche and hardcopy reproductions ERIC makes available *
* via the ERIC Document Reproduction Service (EDRS). EDRS is not *
* responsible for the quality of the original document. Reproductions *
* supplied by EDRS are the best that can be made from the original. *
****

ED127992

FIPS PUB 38

FEDERAL INFORMATION
PROCESSING STANDARDS PUBLICATION

1976 FEBRUARY 15

U.S. DEPARTMENT OF COMMERCE / National Bureau of Standards

U.S. DEPARTMENT OF HEALTH,
EDUCATION & WELFARE
NATIONAL INSTITUTE OF
EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY.

Guidelines

FOR DOCUMENTATION OF COMPUTER PROGRAMS AND AUTOMATED DATA SYSTEMS

CATEGORY: SOFTWARE

SUBCATEGORY: DOCUMENTATION

2

IR003761

U.S. DEPARTMENT OF COMMERCE • Elliot L. Richardson, *Secretary*
James A. Baker, III, *Under Secretary*
Dr. Betsy Ancker-Johnson, *Assistant Secretary* for Science and Technology

NATIONAL BUREAU OF STANDARDS • Ernest Ambler, *Acting Director*

Foreword

The Federal Information Processing Standards Publication Series of the National Bureau of Standards is the official publication relating to standards adopted and promulgated under the provisions of Public Law 89-306 (Brooks Bill) and under Part 6 of Title 15, Code of Federal Regulations. These legislative and executive mandates have given the Secretary of Commerce important responsibilities for improving the utilization and management of computers and automatic data processing systems in the Federal Government. To carry out the Secretary's responsibilities, the NBS, through its Institute for Computer Sciences and Technology, provides leadership, technical guidance, and coordination of government efforts in the development of technical guidelines and standards in these areas.

In October 1974, the Comptroller General of the United States in a report to the Congress noted that "adequate documentation of computer programs is clearly an essential element of efficient and economical use of computer systems." Good documentation should provide information to support the effective management of ADP resources and to facilitate the interchange of information. The NBS is pleased to make these Guidelines for Documentation of Computer Programs and Automated Data Systems available for use by Federal agencies in establishing and evaluating documentation practices.

RUTH M. DAVIS, Director
Institute for Computer Sciences
and Technology

Abstract

These guidelines provide a basis for determining the content and extent of documentation for computer programs and automated data systems. Software development phases and related document types are identified, several examples of documentation options are given, and content guidelines for ten document types are provided. The ten document types are:

Functional Requirements Document
Data Requirements Document
System/Subsystem Specification
Program Specification
Data Base Specification
Users Manual
Operations Manual
Program Maintenance Manual
Test Plan
Test Analysis Report

The guidelines are intended to be a basic reference and a checklist for general use throughout the Federal Government to plan and evaluate documentation practices.

Key words: Automated data systems; computer programs; documentation; documentation content guidelines; FIPS guidelines; software.

Nat. Bur. Stand. (U.S.), Fed. Info. Process. Stand. Publ. (FIPS PUB) 38, 50 pages (1975) CODEN: FIPPAT

For sale by the Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402. (Order by SD Catalog No. C13.52:38.) (Stock No. 003-003-01580-6.) GPO price $1.35.

Federal Information Processing Standards Publication 38

1976 February 15

ANNOUNCING THE

GUIDELINES FOR DOCUMENTATION OF COMPUTER PROGRAMS AND AUTOMATED DATA SYSTEMS

Federal Information Processing Publications are issued by the National Bureau of Standards pursuant to the Federal Property and Administrative Services Act of 1949 as amended, Public Law 89-306 (79 Stat. 1127), and as implemented by Executive Order 11717 (38 FR 12315, dated May 11, 1973), and Part 6 of Title 15 CFR (Code of Federal Regulations).

Name of Guideline. Guidelines for Documentation of Computer Programs and Automated Data Systems.

Category of Guideline. Federal General Applications and Data Standard—Software, Documentation.

Explanation. These guidelines provide a basis for determining the content and extent of documentation for computer programs and automated data systems.

Approving Authority. Department of Commerce, National Bureau of Standards (Institute for Computer Sciences and Technology).

Maintenance Agency. Department of Commerce, National Bureau of Standards (Institute for Computer Sciences and Technology).

Applicability. These guidelines are intended to be a basic reference and a checklist for general use throughout the Federal Government to plan and evaluate documentation practices.

Implementation Schedule. Implementation is desirable at the earliest possible date to achieve more effective use of ADP resources and to facilitate interchange of information about computer programs and automated data systems.

Where documentation standards are already in existence, it is recommended that they be reviewed for conformance with the intent of this guideline and revised as needed to be consistent with the best use of available resources.

Specifications. The following pages define software development phases and related document types, give several examples of documentation options, and provide content guidelines for ten document types.

References.

a. Automated Data System Documentation Standards Manual, Department of Defense Manual 4120.17-M, December 1972.

b. Computer Program Documentation Guideline, National Aeronautics and Space Administration, NHB-2411.1, July 1971.

c. Software Summary for Describing Computer Programs and Automated Data Systems,

Federal Information Processing Standards Publication 30, U.S. Department of Commerce, June 30, 1974.

Definitions.

a. Computer program. A series of instructions or statements, in a form acceptable to a computer, prepared in order to achieve a certain result.

b. Automated data system. A set of logically related computer programs designed to accomplish specific objectives or functions.

Where To Obtain Copies of the Guideline.

a. Federal Government activities should obtain copies of this publication from the established sources within each agency. When there is no established source, purchase orders should be submitted to the National Bureau of Standards, Institute for Computer Sciences and Technology, Office of ADP Standards Management, Technology Building, Washington, D.C. 20234. Refer to the Federal Information Processing Standard Number 38.

b. Others may obtain copies of the FIPS PUB from the Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402 (SD Catalog Number C13.52:38). There is a 25 percent discount on quantities of 100 or more. When ordering, specify document number, title, and SD Catalog Number. Payment may be made by check, money order, coupons, or deposit account.

c. Copies of this FIPS PUB are also available in Microfiche from the National Technical Information Service, 5285 Port Royal Road, Springfield, Virginia 22161 at 95 cents a copy. Refer to Report Number NBS-FIPS-PUB-38.

Acknowledgement

FIPS Task Force 14, Documentation for Information Processing Systems, prepared these guidelines from existing guidelines and practices in Federal agencies and other institutions. Particular recognition is due the following members of the task group:

Fred Cole, Public Health Service (DHEW)
Philip Diamond, Veterans Administration
C. L. Eadie, General Services Administration
James Gillespie, Department of the Navy (DoD)
Robert Hegland, Department of the Navy (DoD)
Helena Hood, General Services Administration
Mitchell Krasny, National Technical Information Service (DOC)
Thomas Kurihara, Department of Agriculture
Beatrice Marron, National Bureau of Standards (DOC)
Carl Palmer, U.S. General Accounting Office
Kenneth Rodey, National Security Agency (DoD)
Joyce Rosencranz, National Security Agency (DoD)
Mark Silverman, Department of the Interior
Roy Young, Department of Health, Education and Welfare

GUIDELINES FOR DOCUMENTATION OF COMPUTER PROGRAMS AND AUTOMATED DATA SYSTEMS

Contents

	Page
INTRODUCTION	5
PART 1. DOCUMENTATION WITHIN THE SOFTWARE LIFE CYCLE	6
1.1. Scope	6
1.2. Phases	6
1.2.1. Initiation	6
1.2.2. Development	6
1.2.3. Operation	6
1.3. Stages	6
1.3.1. Definition	7
1.3.2. Design	7
1.3.3. Programming	7
1.3.4. Test	7
1.4. Document Types	7
1.4.1. Functional Requirements Document	7
1.4.2. Data Requirements Document	7
1.4.3. System/Subsystem Specification	7
1.4.4. Program Specification	7
1.4.5. Data Base Specification	7
1.4.6. Users Manual	7
1.4.7. Operations Manual	7
1.4.8. Program Maintenance Manual	7
1.4.9. Test Plan	7
1.4.10. Test Analysis Report	7
PART 2. DOCUMENTATION CONSIDERATIONS	8
2.1. Responsibilities	8
2.2. Document Audiences	8
2.3. Redundancy	8
2.4. Flexibility	8
2.4.1. "Sizing" of Document Types	8
2.4.2. Combining and Expanding Document Types	8
2.4.3. Format	9
2.4.4. Sequencing of Contents	9
2.4.5. Documenting Multiple Programs or Multiple Files	9
2.4.6. Section/Paragraph Titles	9
2.4.7. Expansion of Paragraphs	9
2.4.8. Flowcharts/Decision Tables	9
2.4.9. Forms	9
2.5. Examples of Documentation Guidance and Criteria	9
PART 3. CONTENT GUIDELINES FOR DOCUMENT TYPES	12
3.1. Functional Requirements Document	15
3.2. Data Requirements Document	21
3.3. System/Subsystem Specification	25
3.4 Program Specification	29
3.5. Data Base Specification	33
3.6. Users Manual	37

Page

3.7. Operations Manual 41
3.8. Program Maintenance Manual 45
3.9. Test Plan 49
3.10. Test Analysis Report 53

Figures

1 Documentation Within the Software Life Cycle 6
2 Cost and/or Usage Threshold Criteria for Extent and Formality 10
3 Total Weighted Documentation Criteria vs Required Document Types 11
4 An Example of Weighting for Twelve Documentation Criteria 12

Introduction

The planning, design, development, and implementation of computer programs and automated data systems [1] represent a considerable investment of human and automated resources. To maximize the return on this investment, and to provide for cost-effective operation, revision, and maintenance, sufficient documentation is needed at each stage of the software development life cycle. This publication has been prepared in response to that need.

Documentation provides information to support the effective management of ADP resources and to facilitate the interchange of information. It serves to:

—Provide managers with technical documents to review at the significant development milestones, to determine that requirements have been met and that resources should continue to be expended.
—Record technical information to allow coordination of later development and use/modification of the software.
—Facilitate understanding among managers, developers, programmers, operators, and users by providing information about maintenance, training, changes, and operation of the software.
—Inform other potential users of the functions and capabilities of the software, so that they can determine whether it will serve their needs.

The quality and consistency of software documentation depend on management commitment and the technical environment. The criteria for evaluating the adequacy of documentation will vary directly with the perceived need for documentation. The utility, quality, and acceptability of the documents prepared will provide a measure of the management judgment exercised in implementing the documentation guidelines.

This publication provides guidelines for the content of software documentation and examples of how management might determine when and how to utilize the ten document types described. Part 1 states the purpose of each document type and its relationship to the software life cycle. Part 2 discusses considerations in using these documentation guidelines including examples of agency or organization level guidance criteria that can be applied to determine the extent of documentation required. Part 3 presents the content guidelines for the ten document types.[2]

[1] Throughout this FIPS PUB 38 "software" is used in lieu of "computer program and/or automated data system."
[2] Note that the Software Summary for Describing Computer Programs and Automated Data Systems (FIPS PUB 30) is considered a component of documentation in this context.

PART 1. DOCUMENTATION WITHIN THE SOFTWARE LIFE CYCLE

1.1. Scope. Computer programs and automated data systems evolve in phases from the time that an idea to create the software occurs through the time that that software produces the required output. It is recognized that there are in current usage many different terminologies to identify these phases and the stages within these phases. Three phases applicable to the software life cycle are: initiation, development, and operation. The development phase is further subdivided into four stages.

This publication provides content guidelines for ten document types generally prepared during the development phase. Figure 1 relates the preparation of the ten document types to the stages in the development phase. The amount of documentation produced is flexible, and this flexibility is discussed in Part 2. Content guidelines for the ten document types is provided in Part 3. Each of these document types can stand alone or be combined with others to meet specific documentation requirements.

FIGURE 1. *Documentation within the software life cycle*

INITIATION PHASE	DEVELOPMENT PHASE				OPERATION PHASE
	Definition Stage	Design Stage	Programming Stage	Test Stage	
	Functional Requirements Document	System/ Subsystem Specification	User Manual		
		Program Specification	Operations Manual		
	Data Requirements Document	Data Base Specification	Program Maintenance Manual		
			Test Plan	Test Analysis Report	

1.2. Phases. While the terminology used to describe the phases is arbitrary, it provides a convenient framework within which the development of software may be discussed.

1.2.1. Initiation. During the Initiation Phase, the objectives and general definition of the requirements for the software are established. Feasibility studies, cost-benefit analyses, and the documentation prepared within this phase are determined by agency procedures and practices.

1.2.2. Development. During the Development Phase, the requirements for the software are determined and the software is then defined, specified, programmed, and tested. Documentation is prepared within this phase to provide an adequate record of the technical information developed.

1.2.3. Operation. During the Operation Phase, the software is maintained, evaluated, and changed as additional requirements are identified.

1.3. Stages. While the terminology used to describe the stages is arbitrary, it provides a convenient framework within which the development of the ten document types may be discussed. It is recognized that not all of the document types are required to document software in every case and that in some cases the various document types may need to be combined. The flexible nature of these guidelines is discussed in Part 2.

1.3.1. Definition. During the definition stage, the requirements for the software and documentation are determined. The Functional Requirements Document and the Data Requirements Document may be prepared.

1.3.2. Design. During the design stage, the design alternatives, specific requirements, and functions to be performed are analyzed and a design is specified. Documents which may be prepared include the System/Subsystem Specification, Program Specification, Data Base Specification, and Test Plan.

1.3.3. Programming. During the programming stage, the software is coded and debugged. Documents which may be prepared during this stage include the Users Manual, Operations Manual, Program Maintenance Manual, and Test Plan.

1.3.4. Test. During the test stage, the software is tested and related documentation reviewed. The software and documentation are evaluated in terms of readiness for implementation. The Test Analysis Report may be prepared.

1.4 Document Types. The purpose of each of the ten document types, described in further detail in part 3, is defined in the following paragraphs.

1.4.1. Functional Requirements Document. The purpose of the Functional Requirements Document is to provide a basis for the mutual understanding between users and designers of the initial definition of the software, including the requirements, operating environment, and development plan.

1.4.2. Data Requirements Document. The purpose of the Data Requirements Document is to provide, during the definition stage of software development, a data description and technical information about data collection requirements.

1.4.3. System/Subsystem Specification. The purpose of the System/Subsystem Specification is to specify for analysts and programmers the requirements, operating environment, design characteristics, and program specifications (if desired) for a system or subsystem.

1.4.4. Program Specification. The purpose of the Program Specification is to specify for programmers the requirements, operating environment, and design characteristics of a computer program.

1.4.5. Data Base Specification. The purpose of the Data Base Specification is to specify the identification, logical characteristics, and physical characteristics of a particular data base.

1.4.6. Users Manual. The purpose of the Users Manual is to sufficiently describe the functions performed by the software in non-ADP terminology, such that the user organization can determine its applicability and when and how to use it. It should serve as a reference document for preparation of input data and parameters and for interpretation of results.

1.4.7. Operations Manual. The purpose of the Operations Manual is to provide computer operation personnel with a description of the software and of the operational environment so that the software can be run.

1.4.8. Program Maintenance Manual. The purpose of the Program Maintenance Manual is to provide the maintenance programmer with the information necessary to understand the programs, their operating environment, and their maintenance procedures.

1.4.9. Test Plan. The purpose of the Test Plan is to provide a plan for the testing of software; detailed specifications, descriptions, and procedures for all tests; and test data reduction and evaluation criteria.

1.4.10. Test Analysis Report. The purpose of the Test Analysis Report is to document the test analysis results and findings, present the demonstrated capabilities and deficiencies for review, and provide a basis for preparing a statement of software readiness for implementation.

PART 2. DOCUMENTATION CONSIDERATIONS

Documentation preparation should be treated as a continuing effort, evolving from preliminary drafts, through changes and reviews, to the documentation and software delivered. The extent of documentation to be prepared is a function of agency management practices and the size, complexity and risk of the project.

2.1. Responsibilities. Separable responsibilities which are inherent in the flexible nature of these guidelines are:

a. Definition of agency guidance to project managers as to what documentation should be prepared under various conditions and, perhaps, to what levels of extent, detail, and formality. See Examples A and B in paragraph 2.5.

b. Determination by a project manager of the documentation plan for a specific project, including:

(1) What document types apply and should be prepared.
(2) The formality, extent, and detail of the documentation.
(3) Responsibilities and a schedule of preparation for the documentation.
(4) Procedures and schedule of review, approval, and distribution and the distribution list.
(5) Responsibilities for documentation maintenance and change control through the development phase.

The formality, extent, and level of detail, and other determinations by the project manager specific cases will be more consistent if agency guidance and criteria are established. In gener as the size, complexity, and risk of a project increase, so does the need for formality, extent, and level of detail of the documentation. The Users, Operations, and Program Maintenance Manuals should be formal since they support the use of the software, particularly if the software will be used outside of the developing organization or if extensive changes are expected during the life of the software.

2.2 Document Audiences. Each document type is written for a particular "audience." The audience may be an individual or a group of individuals who are expected to use the document contents to perform a function, e.g., operation, maintenance, design, programming. The information should be presented using the terminology and level of detail appropriate to the audience.

2.3. Redundancy. The ten document types in this guideline have some apparent redundancy. This apparent redundancy is of two types. Introductory material has been included in each document type to provide the reader with a frame of reference. This information has been included to provide the "stand alone" approach, and understanding of the document with a minimum need for cross-referencing to parts of other documents that may have been produced. A second type of apparent redundancy is that most document types specify, for example, descriptions of inputs, outputs, and equipment to be included. The information that should be included in each of the document types, differs in context and, perhaps, in terminology and level of detail, since the information is intended to be read by different audiences and at different points in the software life cycle.

2.4. Flexibility. Flexibility in the use of the document content guidelines is provided by the basic organization of contents. An attempt has been made to provide an internally consistent organization scheme. The following paragraphs describe various options which should be considered.

2.4.1. "Sizing" of Document Types. Each document type outline may be used to prepare documents that range from a few to several hundred pages in length. The size depends on the size and complexity of the project and the judgment of the project manager as to the level of detail necessary for the environment in which the software will be developed or run.

2.4.2. Combining and Expanding Document Types. It is occasionally necessary to combine several document types under one cover or to produce several volumes of the same document type. Document types that can be combined into one are, for example, the Users, Operations, and Program Maintenance Manuals. When this is done, the substance of the contents covered

by each document type should be presented using the outline of that document type, for example, Part I–Users, Part II–Operations, and Part III–Program Maintenance.

When a system is extremely large or is to be documented in a modular fashion, a document may be prepared for each module. In some cases, the size of a document may necessitate that it be issued in multiple volumes to allow ease of user reference. In such cases, the document should be separated at a section division. The contents of the Test Plan document type, for example, may be separated between the sections of plan, specifications and evaluation, and specific test descriptions.

2.4.3. Format. The content guidelines in Part 3 have been prepared using a generally consistent format. Use of this particular format is encouraged but is not essential. It is a tested and accepted format.

2.4.4. Sequencing of Contents. In general, the order of the sections and paragraphs in a particular document type should be the same as shown in the content guidelines in Part 3. The order may be changed if it significantly enhances the presentation.

2.4.5. Documenting Multiple Programs or Multiple Files. Many of the document type content outlines anticipate and are adaptable to documenting a system and its subsystems, multiple programs, or multiple files. All of these outlines can, of course, be used for a single system, subsystem, program, data base, or file.

2.4.6. Section/Paragraph Titles. In general, the titles of sections and paragraphs should be the same as shown in the content guidelines. The titles may be modified to reflect terminology unique to the software being documented if the change significantly enhances the presentation. Sections or paragraphs may be added or deleted as local requirements dictate.

2.4.7. Expansion of Paragraphs. Many of the document types have paragraphs with a general title and a list of factors that might be discussed within that paragraph. The intent of the content guidelines is not to prescribe a discussion of each of these items, but to suggest that these items be considered in writing that paragraph. These and all other paragraphs may be expanded and further subdivided to enhance the presentation.

2.4.8. Flowcharts/Decision Tables. The graphic representations of some problem solutions are treated best in the form of flowcharts, others in the form of decision tables. Either may be included in or appended to the documents produced.

2.4.9. Forms. The use of specific forms is dependent on practices in an agency. Some of the information specified in a paragraph in the content guidelines may be recorded on such forms. If so, the form can be referenced from the appropriate paragraph. The use of standard forms is encouraged.

2.5. Examples of Documentation Guidance and Criteria. The formality, extent, and level of detail of documentation to be prepared is a function of agency ADP management practices and the size, complexity, and risk of a project. The following examples were taken from two Federal agency directives, but are amended to conform to the naming of document types in this publication. The examples illustrate how criteria could be established to aid project managers in determining the extent and level of detail of documentation required.

Example A presents a scheme using development cost and document audience as two criteria to establish thresholds for documentation requirements. See the following pages and Figure 2.

Example B presents a scheme using twelve criteria with weighting factors and a scale of the total weighted criteria to establish formal documentation requirements. Figure 3 illustrates the application of the weighted criteria shown in Figure 4. The procedure to use these tables is:

1. Weight the software by each of the twelve criteria in Figure 4.
2. Sum the weights assigned. (Total weighted criteria.)
3. Find the row in Figure 3 that lists the document types to be prepared.

200-393 O - 76 - 2

FIGURE 2. EXAMPLE A. *Cost and/or usage threshold criteria for extent and formality*

Level	If PROJECT COST:	Or USAGE	Then DOCUMENTATION ELEMENTS	And EXTENT OF EFFORT
1	Less than $1000 Or One Man-month	One Shot (Single Use)	Software Summary plus any incidentally produced documentation.	No special effort, normal good practice.
2	$1000 to $5000	Special or Limited Purpose or Application	Level 1 plus Users Manual and Operations Manual.	Minimal documentation effort, spent on informal documentation. No formal documentation effort.
3	Over $5000	Multipurposed, or Multiuser	Level 2 plus Functional Requirements Document, Program Specification, Program Maintenance Manual, Test Plan, Test Analysis Report, and System/Subsystem Specification.	All basic elements of documentation should be typewritten, but need not be prepared in finished format for publication or require external edit or review.
4	Over $5000	Publicly Announced, or Critical to Operations	Level 3 produced in a form suitable for publication.	At a minimum, all basic elements prepared for formal publication, including external review and edit.

EXAMPLE A. LEVELS OF DOCUMENTATION

DEFINITIONS OF LEVELS

To protect against both over and under documentation, computer program documentation has been divided into four levels. From lowest to highest these levels of documentation are: (1) minimal level, (2) internal level, (3) working document [3] level, and (4) formal publication level. The criteria determining these levels of documentation are described in the following paragraphs, and summarized in Figure 2. Additional criteria peculiar to an installation and/or judgment relative to program sharing potential, life expectancy, and usage frequency are also appropriate factors to be considered in the determination of documentation levels.

MINIMAL LEVEL (LEVEL 1)

Level 1 documentation guidelines are applicable to single use programs, or one-shot jobs, of minimal complexity. Although no significant documentation cost should be added, there exists the requirement to show what type of work is being produced and what a given program really does. Hence, it is desirable to keep on file for a minimum period of time the documentation which results from the development of the programs, i.e., program abstract, compile listing, test cases, etc. The criteria for categorizing a program as Level 1 can be its expected usage or the resource expended in its generation, in man-hours or dollars, and may be modified for the peculiar requirements of the installations. Suggested resource expenditure criteria are programs requiring less than one man-month effort or less than $1,000 (these are not assumed to be equal).

INTERNAL LEVEL (LEVEL 2)

Level 2 documentation applies to special purpose programs which, after careful consideration of the possible interest of others, appear to have no sharing potential and to be designed for use only by the requesting scientist or manager in an environment over which he has cognizance. Large programs which have a short life expectancy also fall into this level. The documentation required (other than Level 1) is that necessary for deck setup and modifications. This requirement can be satisfied by the inclusion of detail input/output formats, setup instructions, and the liberal use of comment cards in the source deck to provide clarification in the compile listing. In summary, the effort spent toward formal documentation for Level 2 programs should be minimal.

[3] The term "working document" or "working paper" as used in this guideline refer to typewritten documents, not necessarily prepared in finished format suitable for publication nor subject to external editorial review.

WORKING DOCUMENT LEVEL (LEVEL 3)

This level applies to programs which are expected to be used by a number of people in the same installation or which may be transmitted on request to other installations or to contractors or grantees. The format of the documentation at this level should include, as a minimum, all elements of documentation. All basic elements of documentation should be prepared in typewritten form, but not necessarily in a finished format suitable for publication. Normally, it will not be formally reviewed or edited above the review required for a working paper. However, if there are certain programs important to the activities of the installations, but not considered appropriate for publication, then local more stringent documentation review standards should be applied.

FORMAL PUBLICATION LEVEL (LEVEL 4)

This level applies to programs which are of sufficient general interest and value to be announced outside the originating installation. This level of documentation is also desirable if the program is to be referenced by a scientific publication or paper. The format of the documentation at this level should comply with the guidelines on elements of documentation suitable for inclusion in one of the scientific and technical publication series with the attendant review and editing procedures.

Also considered to be within this level are those programs which are critical to the activities of the installation. These programs should be documented in a formal, rigorous manner, with in-depth review and special configuration control procedures enforced. Recurring management applications, such as payroll, should be considered for inclusion in this category so as to maintain an accurate history of conformation to changing laws, rules, and regulations.

FIGURE 3. EXAMPLE B. *Total weighted documentation criteria vs required document types*

(See Figure 4 to determine total weighted criteria.)

TOTAL WEIGHTED CRITERIA	Software Summary	Users Manual	Operations Manual	Program Maintenance Manual	Test Plan	Functional Requirements Document	System/Subsystem Specification	Test Analysis Report	Program Specification	Data Requirements Document	Data Base Specification
0-12*	X										
12-15*	X	X									
12-26	X	X	X	X	X			**		***	***
24-38	X	X	X	X	X	X		**		***	***
36-50	X	X	X	X	X	X	X	X		***	***
48-60	X	X	X	X	X	X	X	X	X	***	***

NOTES: * Additional document types may be required at lower weighted criteria totals to satisfy local requirements.
** The Test Analysis Report logically should be prepared, but may be informal.
*** Preparation of the Data Requirements Document and Data Base Specification is situationally dependent.

FIGURE 4. EXAMPLE B. *An example of weighting for twelve documentation criteria (See Figure 3 for application of total weighted criteria to determination of required documentation types.)*

	WEIGHTS				
Criteria	1	2	3	4	5
1. Originality required	None—reprogram on different equipment	Minimum—more stringent requirements	Limited—new interfaces	Considerable—apply existing state of art to environment	Extensive—require advance in state of the art
2. Degree of generality	Highly restricted. Single purpose	Restricted—parameterized for a range of capacities	Limited flexibility. Allows some change in format	Multi-purpose. Flexible format. Range of subjects	Very flexible—able to handle a broad range of subject matter on different equipment
3. Span of operation	Local or utility	Component command	Single command	Multi-command	Defense Department. World wide.
4. Change in scope and objective	None	Infrequent	Occasional	Frequent	Continuous.
5. Equipment complexity	Single machine. Routine processing	Single machine. Routine processing. Extended peripheral system	Multi-computer. Standard peripheral system	Multi-computer. Advanced programming. Complex peripheral system	Master control system. Multi-computer auto input/output and display equipment.
6. Personnel assigned	1–2	3–5	5–10	10–18	18 and over
7. Developmental cost	1–10k	10–50k	50–200k	200–500k	Over 500k
8. Criticality	Data processing	Routine operations	Personnel safety	Unit survival	National defense
9. Average response time to program change	2 or more weeks	1–2 weeks	3–7 days	1–3 days	1–24 hours
10. Average response time to data inputs	2 or more weeks	1–2 weeks	1–7 days	1–24 hours	0–60 minutes
11. Programming languages	High level language	High level and limited assembly language	High level and extensive assembly language	Assembly language	Machine language
12. Concurrent software development	None	Limited	Moderate	Extensive	Exhaustive

PART 3. CONTENT GUIDELINES FOR DOCUMENT TYPES

Part 3 provides content guidelines for the following ten document types discussed in Parts 1 and 2.

3.1 Functional Requirements Document
3.2 Data Requirements Document
3.3 System/Subsystem Specification
3.4 Program Specification
3.5 Data Base Specification
3.6 Users Manual
3.7 Operations Manual
3.8 Program Maintenance Manual
3.9 Test Plan
3.10 Test Analysis Report

The document types are presented in the order of development within the software life cycle. Included for each document type are a table of contents and a description of the contents of that document type. The page numbers given in the table of contents for each document type are those within the boxes.

The purpose of the Functional Requirements Document is to provide a basis for the mutual understanding between users and designers of the initial definition of the software, including the requirements, operating environment, and development plan.

Contents

	Page
SECTION 1. GENERAL INFORMATION	2
1.1. Summary	2
1.2. Environment	2
1.3. References	2
SECTION 2. OVERVIEW	2
2.1. Background	2
2.2. Objectives	2
2.3. Existing Methods and Procedures	2
2.4. Proposed Methods and Procedures	2
2.5. Summary of Improvements	3
2.6. Summary of Impacts	3
2.6.1. Equipment Impacts	3
2.6.2. Software Impacts	3
2.6.3. Organizational Impacts	3
2.6.4. Operational Impacts	3
2.6.5. Development Impacts	3
2.7. Cost Considerations	3
2.8. Alternative Proposals	3
SECTION 3. REQUIREMENTS	4
3.1. Functions	4
3.2. Performance	4
3.2.1. Accuracy	4
3.2.2. Validation	4
3.2.3. Timing	4
3.2.4. Flexibility	4
3.3. Inputs-Outputs	4
3.4. Data Characteristics	4
3.5. Failure Contingencies	4
SECTION 4. OPERATING ENVIRONMENT	5
4.1. Equipment	5
4.2. Support Software	5
4.3. Interfaces	5
4.4. Security and Privacy	5
4.5. Controls	5
SECTION 5. DEVELOPMENT PLAN	5

Functional Requirements Document

1. GENERAL INFORMATION

1.1. Summary. Summarize the general nature of the software to be developed.

1.2. Environment. Identify the project sponsor, developer, user, and computer center or network where the software is to be implemented.

1.3. References. List applicable references, such as:

a. Project request (authorizations).
b. Previously published documents on the project.
c. Documentation concerning related projects.
d. FIPS publications and other reference documents.

2. OVERVIEW

2.1. Background. Present the purpose and scope of the software, and any background information that would orient the reader. Explain relationships with other software.

2.2. Objectives. State the major performance objectives of the software, including examples. Identify anticipated operational changes that will affect the software and its use.

2.3. Existing Methods and Procedures. Describe the current methods and procedures that satisfy the existing objectives. Include information on:

a. Organizational and personnel responsibilities.
b. Equipment available and required.
c. Volume and frequency of inputs and outputs.
d. Deficiencies and limitations.
e. Pertinent cost considerations.

Illustrate the existing data flow from data acquisition through its processing and eventual output. Explain the sequence in which operational functions are performed by the user.

2.4. Proposed Methods and Procedures. Describe the proposed software and its capabilities. Identify techniques and procedures from other software that will be used or that will become part of the proposed software. Identify the requirements that will be satisfied by the proposed software. Include information on:

a. Organizational and personnel responsibilities.
b. Equipment available and required.
c. Volume and frequency of inputs and outputs.
d. Deficiencies and limitations.
e. Pertinent cost considerations (developmental as well as operational).

Illustrate the proposed data flow to present an overall view of the planned capabilities. Describe any capabilities in the existing software that may be changed by the proposed software. State the reasons for these changes. Explain the sequence in which operational functions are to be performed by the user.

2.5. **Summary of Improvements.** Itemize improvements to be obtained from the proposed software, such as:

a. New capabilities.
b. Upgraded existing capabilities.
c. Elimination of existing deficiencies.
d. Improved timeliness, e.g., decreased response time or processing time.
e. Elimination or reduction of existing capabilities that are no longer needed.

2.6. **Summary of Impacts.** Summarize the anticipated impacts of the proposed software on the present system, in the following categories:

2.6.1. Equipment Impacts. Summarize changes to currently available equipment, as well as new equipment requirements and building modifications.

2.6.2. Software Impacts. Summarize any additions or modifications needed to existing applications and support software in order to adapt them to the proposed software.

2.6.3. Organizational Impacts. Summarize organizational impacts, such as:

a. Functional reorganization.
b. Increase/decrease in staff level.
c. Upgrade/downgrade of staff skills.

2.6.4. Operational Impacts. Summarize operational impacts, such as modifications to:

a. Staff and operational procedures.
b. Relationships between the operating center and the users.
c. Procedures of the operating center.
d. Data (sources, volume, medium, timeliness).
e. Data retention and retrieval procedures.
f. Reporting methods.
g. System failure consequences and recovery procedures.
h. Data input procedures.
i. Computer processing time requirements.

2.6.5. Developmental Impacts. Summarize developmental impacts, such as:

a. Specific activities to be performed by the user in support of development of the proposed software.
b. Resources required to develop the data base.
c. Computer processing resources required to develop and test the new software.

2.7. **Cost Considerations.** Describe resource and cost factors that may influence the development, design, and continued operation of the proposed software. Discuss other factors which may determine requirements, such as interfaces with other automated systems and telecommunication facilities.

2.8. **Alternative Proposals.** If alternative software has been proposed to satisfy the requirements, describe each alternative. Compare and contrast the alternatives. Explain the selection reasoning.

3. REQUIREMENTS

3.1 **Functions.** State the functions required of the software in quantitative and qualitative terms, and how these functions will satisfy the performance objectives.

3.2. **Performance.** Specify the performance requirements.

3.2.1. Accuracy. Describe the data accuracy requirements imposed on the software, such as:

a. Mathematical.
b. Logical.
c. Legal.
d. Transmission.

3.2.2. Validation. Describe the data validation requirements imposed on the software.

3.2.3. Timing. Describe the timing requirements imposed on the software, such as, under varying conditions:

a. Response time.
b. Update processing time.
c. Data transfer and transmission time.
d. Throughput time.

3.2.4. Flexibility. Describe the capability for adapting to changes in requirements, such as:

a. Changes in modes of operation.
b. Operating environment.
c. Interfaces with other software.
d. Accuracy and validation timing.
e. Planned changes or improvements.

Identify the software components which are specifically designed to provide this flexibility.

3.3. **Inputs-Outputs.** Explain and show examples of the various data inputs. Specify the medium (disk, cards, magnetic tape), format, range of values, accuracy, etc. Provide examples and explanation of the data outputs required of the software, and any quality control outputs that have been identified. Include descriptions or examples of hard copy reports (routine, situational and exception) as well as graphic or display reports.

3.4. **Data Characteristics.** Describe individual and composite data elements by name, their related coded representations, as well as relevant dictionaries, tables, and reference files. Estimate total storage requirements for the data and related components based on expected growth.

3.5. **Failure Contingencies.** Specify the possible failures of the hardware or software, the consequences (in terms of performance), and the alternative courses of action that may be taken to satisfy the information requirements. Include:

a. Back-up. Specify back-up techniques, i.e., the redundancy available in the event the primary system element goes down. For example, a back-up technique for a disk medium would be to record periodically the contents of the disk to a tape.
b. Fallback. Explain the fallback techniques, i.e., the use of another system or other means to accomplish some portion of requirements. For example, the fallback technique for an automated system might be manual manipulation and recording of data.
c. Recovery and Restart. Discuss the recovery and restart techniques, i.e., the capability to resume execution of software from a point in the software subsequent to which a hardware or software problem occurred, or the re-running of the software from the beginning.

4. OPERATING ENVIRONMENT

4.1. Equipment. Identify the equipment required for the operation of the software. Identify any new equipment required and relate it to specific functions and requirements to be supported. Include information such as:

a. Processor and size of internal storage.
b. Storage, online and offline, media, form, and devices.
c. Input/output devices, online and offline.
d. Data transmission devices.

4.2. Support Software. Identify the support software and describe any test software. If the operation of the software depends on changes to support software, identify the nature and planned date of these changes.

4.3. Interfaces. Describe the interfaces with other software.

4.4. Security and Privacy. Describe the overall security and privacy requirements imposed on the software. If no specific requirements are imposed, state this fact.

4.5. Controls. Describe the operational controls imposed on the software. Identify the sources of these controls.

5. DEVELOPMENT PLAN

Discuss in this section the overall management approach to the development and implementation of the proposed software. Include a list of the documentation to be produced, time frames and milestones for the development of the software, and necessary participation by other organizations to assure successful development.

The purpose of the Data Requirements Document is to provide, during the definition stage of software development, a data description and technical information about data collection requirements.

Contents

	Page
SECTION 1. GENERAL INFORMATION	2
1.1. Summary	2
1.2. Environment	2
1.3. References	2
1.4. Modification of Data Requirements	2
SECTION 2. DATA DESCRIPTION	2
2.1. Static Data	2
2.2. Dynamic Input Data	2
2.3. Dynamic Output Data	2
2.4. Internally Generated Data	2
2.5. Data Constraints	2
SECTION 3. DATA COLLECTION	3
3.1. Requirements and Scope	3
3.2 Input Responsibilities	3
3.3. Procedures	3
3.4. Impacts	3

Data Requirements Document

1. GENERAL INFORMATION

1.1. Summary. Summarize the general nature of the software for which these data requirements are being defined.

1.2. Environment. Identify the project sponsor, developer, user organization, and computer center where the software is to be installed. Show the relationships of these data requirements and those of other software.

1.3. References. List applicable references, such as:

a. Project request (authorization).
b. Previously published documents on the project.
c. Documentation concerning related projects.
d. FIPS publications and other reference documents.

1.4. Modification of Data Requirements. Describe or reference procedures for implementing and documenting changes to these data requirements.

2. DATA DESCRIPTION

Separate the data description into two categories, static data and dynamic data. Static data is defined as that data which is used mainly for reference during operation and is usually generated or updated in widely separated time frames independent of normal runs. Dynamic data includes all data which is intended to be updated and which is input during a normal run or is output. Arrange the data elements in each category in logical groupings, such as functions, subjects, or other groupings which are most relevant to their use.

2.1. Static Data. List the static data elements used for either control or reference purposes.

2.2. Dynamic Input Data. List the dynamic input data elements which constitute the data intended to be changed by a normal run or during online operation.

2.3. Dynamic Output Data. List the dynamic output data elements which constitute the data intended to be changed by a normal run or during online operation.

2.4. Internally Generated Data. List the internally generated data of informational value to the user or developer.

2.5. Data Contraints. State the constraints on the data requirements. Indicate the limits of the data requirements with regard to further expansion or utilization, such as the maximum size and number of files, records, and data elements. Emphasize the constraints that could prove critical during design and development.

3. DATA COLLECTION

3.1. Requirements and Scope. Describe the type of information required to document the characteristics of each data element. Specify information to be collected by the user and that to be collected by the developer. It should be logically grouped and presented. Include:

a. Source of Input. Identify the source from which the data will be entered, e.g., an operator, station, organizational unit, or its component group.
b. Input Medium and Device. Identify the medium and hardware device intended for entering the data into the system. In those cases where only certain special stations are to be legitimate entry points, they should be specified.
c. Recipients. Identify the intended recipients of the output data.
d. Output Medium and Device. Identify the medium and hardware device intended for presenting output data to the recipient. Specify whether the recipient is to receive the data as part of a hard copy printout, a symbol in a CRT display, a line on a drawing, a colored light, an alarm bell, etc. If the output is to be passed to some other automated system, the medium should be described, such as magnetic tape, punched cards, or an electronic signal to a solenoid switch.
e. Critical Value. One value from a ange of values of data may have particular significance to a recipient.
f. Scales of Measurement. Specify for numeric scales, units of measurement, increments, scale zero-point, and range of values. For non-numeric scales, any relationships indicated by the legal values should be stated.
g. Conversion Factors. Specify the conversion factors of measured quantities that must go through analog or digital conversion processes.
h. Frequency of Update and Processing. Specify the expected frequency of data change and the expected frequency of processing input data. If the input arrives in a random or in an "as occurred" manner, both the average frequency and some measure of the variance must be specified.

3.2. Input Responsibilities. Provide recommendations as to responsibilities for preparing specific data inputs. Include any recommendations regarding the establishment of a data input group. Specify by source those data inputs dependent on interfacing software or unrelated organizations.

3.3. Procedures. Provide specific instructions for data collection procedures. Include detailed formats where applicable, and identify expected data communications media and timing of inputs.

3.4. Impacts. Describe the impacts of these data requirements on equipment, software and the user and developer organizations.

The purpose of the System/Subsystem Specification is to specify for Analysts and Programmers the requirements, operating environment, design characteristics, and program specifications (if desired) for a system or subsystem.

Contents

	Page
SECTION 1. GENERAL INFORMATION	2
1.1. Summary	2
1.2 Environment	2
1.3. References	2
SECTION 2. REQUIREMENTS	2
2.1. Description	2
2.2. Functions	2
2.3. Performance	2
2.3.1. Accuracy	2
2.3.2. Validation	2
2.3.3. Timing	2
2.3.4. Flexibility	2
SECTION 3. OPERATING ENVIRONMENT	3
3.1. Equipment	3
3.2. Support Software	3
3.3. Interfaces	3
3.4. Security and Privacy	3
3.5. Controls	3
SECTION 4. DESIGN CHARACTERISTICS	3
4.1. Operations	3
4.2. System/Subsystem Logic	3
SECTION 5. PROGRAM SPECIFICATIONS	3
5.1. Program (Identify) Specification	3
5.N. Program (Identify) Specification	3

System/Subsystem Specification

1. GENERAL INFORMATION

1.1. Summary. Summarize the specifications and functions of the system/subsystem to be developed.

1.2. Environment. Identify the project sponsor, developer, user, and computer center or network on which the system is to be implemented.

1.3. References. List applicable references, such as:

a. Project request (authorizations).
b. Previously published documents on the subject
c. Documentation concerning related projects.
d. FIPS publications and other reference documents.

2. REQUIREMENTS

2.1. Description. Provide a general description of the system/subsystem to establish a frame of reference for the remainder of the document. Include a summary of functional requirements to be satisfied by this system/subsystem. Show the general interrelationship of the system/subsystem components.

2.2. Functions. Specify the system/subsystem functions in quantitative and qualitative terms and how the functions will satisfy the functional requirements.

2.3. Performance. Specify the performance requirements.

2.3.1. Accuracy. Describe the data accuracy requirements imposed on the system or subsystem, such as:

a. Mathematical.
b. Logical.
c. Legal.
d. Transmission.

2.3.2. Validation. Describe the data validation requirements imposed on the system/subsystem.

2.3.3. Timing. Describe the timing requirements imposed on the software, such as, under varying conditions:

a. Response time.
b. Update processing time.
c. Data transfer and transmission time.
d. Throughput time.

2.3.4. Flexibility. Describe the capability for adapting the program to changes in requirements, such as:

The organization of the contents of Sections 2, 3, 4, and 5 may vary according to the purpose of the documentation. See Example following this content guideline, page 28.

a. Changes in modes of operation.
b. Operating environment.
c. Interfaces with other software.
d. Accuracy and validation and timing.
e. Planned changes or improvements.

Identify the system/subsystem components which are specifically designed to provide this flexibility.

3. OPERATING ENVIRONMENT

3.1. Equipment. Identify the equipment required for the operation of the system/subsystem. Identify any new equipment required and relate it to specific functional requirements to be supported. Include information, such as:

a. Processor and size of internal storage.
b. Storage, online and offline, media, form, and devices.
c. Input/output devices, online and offline.
d. Data transmission devices.

3.2. Support Software. Identify the support software and describe any test software. If the operation of the system/subsystems depends on changes to support software, identify the nature and planned date of these changes.

3.3. Interfaces. Describe the interfaces with other software.

3.4. Security and Privacy. Describe the overall security and privacy requirements imposed on the system/subsystem. If no specific requirements are imposed, state this fact.

3.5. Controls. Describe the operational controls imposed on the system/subsystem. Identify the sources of these controls.

4. DESIGN CHARACTERISTICS

4.1. Operations. Describe the operating characteristics of the user and computer centers where the software will be operational.

4.2. System/Subsystem Logic. Describe the logic flow of the entire system/subsystem in the form of a flowchart. The flow should provide an integrated presentation of the system/subsystem dynamics, of entrances and exits, computer programs, support software, controls, and data flow.

5. PROGRAM SPECIFICATIONS

5.1. Program (Identify) Specification. Specify the system/subsystem functions to be satisfied by the computer program.

a. Describe the program requirements.
b. Describe the operating environment.
c. Describe the design characteristics of the program including inputs, program logic, outputs, and data base.

5.N. Program (Identify) Specification. Describe the remaining computer programs in a manner similar to the paragraph above.

EXAMPLES OF ALTERNATIVE SECTION OUTLINES

Sections 2, 3, and 4 of this specification may follow one of several alternative outlines depending on the purpose to which the documentation is directed. Examples of alternative purposes and the corresponding outline are shown below.

Example A: When this document is directed to the documentation of a given system and is not to specifically include the documentation of any subsystem, the appropriate title would be "System Specification." The outline for the specification would be:

REQUIREMENTS
- Description
- Functions
- Performance

OPERATING ENVIRONMENT
- Equipment
- Support Software
- Interfaces
- Security and Privacy
- Controls

DESIGN CHARACTERISTICS
- Operations
- Logic

Example B: When this documents is directed to the documentation of a given subsystem, the appropriate title would be "Subsystem Specification." The outline for the specification would be the same as Example A above.

Example C: When this document is directed to the documentation of a system and its subsystems, the appropriate title would be "System and Subsystem Specifications." The outline, in brief, for the specification would be:

System REQUIREMENTS
System OPERATING ENVIRONMENT
System DESIGN CHARACTERISTICS
Subsystem 1 (Identify)
- REQUIREMENTS
- OPERATING ENVIRONMENT
- DESIGN CHARACTERISTICS
- PROGRAM SPECIFICATIONS

Subsystem 'n' (Identify)

Example D: In any of the above examples, the program specifications may be documented within as a separate section; as subsections to each subsystem section; or may be documented in a separate document, "Program Specification."

The purpose of the Program Specification is to specify for programmers the requirements, operating environment, and design characteristics of a computer program.

Contents

	Page
SECTION 1. GENERAL INFORMATION	2
1.1. Summary	2
1.2. Environment	2
1.3. References	2
SECTION 2. REQUIREMENTS	2
2.1. Program Description	2
2.2. Functions	2
2.3. Performance	2
2.3.1. Accuracy	2
2.3.2. Validation	2
2.3.3. Timing	2
2.3.4. Flexibility	2
SECTION 3. OPERATING ENVIRONMENT	3
3.1. Equipment	3
3.2. Support Software	3
3.3. Interfaces	3
3.4. Storage	3
3.5. Security and Privacy	3
3.6. Controls	3
SECTION 4. DESIGN CHARACTERISTICS	3
4.1. Operating Procedures	3
4.2. Inputs	4
4.3. Program Logic	4
4.4. Outputs	4
4.5. Data Base	4
4.5.1 Logical Characteristics	4
4.5.2 Physical Characteristics	4

Program Specification

1. GENERAL INFORMATION

1.1. **Summary.** Summarize the specifications and functions of the computer program to be developed.

1.2. **Environment.** Identify the project sponsor, developer, user, and computer center where the computer program is to be run.

1.3. **References.** List applicable references, such as:

a. Project request (authorization).
b. Previously published documents on the subject.
c. Documentation concerning related projects.
d. FIPS publications and other reference documents.

2. REQUIREMENTS

2.1. **Program Description.** Provide a general description of the program to establish a frame of reference for the remainder of the document. Include a summary description of the system/subsystem functions to be satisfied by this program.

2.2. **Functions.** Specify the functions of the program to be developed. If the program in itself does not fully satisfy a system/subsystem function, show the relationship to other programs which in aggregate satisfy that function.

2.3. **Performance.** Specify the performance requirements.

2.3.1. Accuracy. Describe data accuracy requirements imposed on the program, such as:

a. Mathematical.
b. Logical.
c. Legal.
d. Transmission.

2.3.2. Validation. Describe the data validation requirements imposed on the program.

2.3.3. Timing. Describe the timing requirements imposed on the program, such as, under varying conditions:

a. Response time.
b. Update processing time.
c. Data transfer and transmission time.
d. Throughput and internal processing time.

2.3.4. Flexibility. Describe the capability for adapting the program to changes in requirements, such as:

a. Modes of operation.
b. Operating environment.
c. Interfaces with other programs.
d. Accuracy, validation, and timing.
e. Planned changes or improvements.

Identify the components of the program which are designed to provide this flexibility.

3. OPERATING ENVIRONMENT

3.1. Equipment. Identify the equipment required for the operation of the program. Include information on equipment required, such as:

a. Processor and size of internal storage.
b. Storage, online and offline, media, form, and devices.
c. Input/Output devices, online and offline, and capacities.
d. Data transmission devices.

3.2. Support Software. Identify the support software and describe any test programs. If the operation of the program depends on changes to support software, identify the nature and planned date of these changes.

3.3. Interfaces. Describe all interactions with the operator. Describe all interactions with other software, including sequence or procedure relationships and data interfaces.

3.4. Storage. Specify the storage requirements and any constraints and conditions.

a. Internal. Describe and illustrate the use of internal storage areas, including indexing and working areas. Briefly state the equipment constraints and design considerations that affect the use of internal storage.
b. Device. List by device type all peripheral storage required. Briefly state any constraints imposed on storage requirements by each storage device. State requirements for permanent and temporary storage, including overlays.
c. Offline. Describe the form, media and storage requirements of all offline storage.

3.5. Security and Privacy. Describe the security and privacy requirements imposed on the program, the inputs, the outputs, and the data bases. If no specific requirements are imposed, state this fact.

3.6. Controls. Describe the program controls such as record counts, accumulated counts, and batch controls. Identify the sources of these controls.

4. DESIGN CHARACTERISTICS

4.1. Operating Procedures. Describe the operating procedures and any special program functions or requirements necessary for its implementation. Describe the load, start, stop, recovery, and restart procedures. Describe all other interactions of the program with the operator.

4.2. **Inputs.** Provide information about the characteristics of each input to the program, such as.

a. Title and tag.
b. Format and type of data, such as a record layout.
c. Validation criteria.
d. Volume and frequency.
e. Means of entry.
f. Source document and its disposition, or specific interface source.
g. Security and privacy conditions.

4.3. **Program Logic.** Describe the program logic. The logical flow should be presented in graphic form (flowcharts, decision logic tables) supplemented by narrative explanations.

4.4. **Outputs.** Provide information about the characteristics of each output from the program, such as:

a. Title and tag.
b. Format specifications, such as a report format.
c. Selection criteria for display, output, or transfer.
d. Volume and frequency.
e. Output media.
f. Description of graphic displays and symbols.
g. Security and privacy conditions.
h. Disposition of products.
i. Description of sequence of displays, display contents, fixed and variable formats, and display of error conditions.

4.5. **Data Base.** Describe the logical and physical characteristics of any data base used by the program.

4.5.1. Logical Characteristics. Describe for each unique set, file, record, element, or item of data, its identification, definition, and relationships.

4.5.2. Physical Characteristics. Describe in terms of this data base, the storage requirements for program data, specific access method, and physical relationships of access (index, device, area), design considerations, and access security mechanisms.

The purpose of the Data Base Specification is to specify the identification, logical characteristics, and physical characteristics of a particular data base.

Contents

	Page
SECTION 1. GENERAL INFORMATION	2
1.1. Summary	2
1.2. Environment	2
1.3. References	2
SECTION 2. DESCRIPTION	2
2.1. Identification	2
2.2. Using Software	2
2.3. Conventions	2
2.4. Special Instructions	2
2.5. Support Software	2
SECTION 3. LOGICAL CHARACTERISTICS	3
SECTION 4. PHYSICAL CHARACTERISTICS	3
4.1. Storage	3
4.2. Access	3
4.3. Design Considerations	3

Data Base Specification

1. GENERAL INFORMATION

1.1. Summary. Summarize the purpose of the data base and general functions of the using software.

1.2. Environment. Identify the project sponsor, developer, user organization, and computer center where the software and data base are to be installed.

1.3. References. List applicable references, such as:

a. Project request (authorization).
b. Previously published documents on the project.
c. Documentation concerning related projects.
d. FIPS publications and other reference documents.

2. DESCRIPTION

2.1. Identification. Specify the code name, tag, or label by which the data base is to be identified. If the data base is to be experimental, test, or temporary, specify this characteristic and effective dates or period. Any additional identification information should also be given.

2.2. Using Software. Identify all software intended to use or access this data base. Identify for each: the software name, code name, and any release or version number.

2.3. Conventions. Describe all labeling or tagging conventions essential for a programmer or analyst to use this data base specification.

2.4. Special Instructions. Provide any special instructions to personnel who will contribute to the generation of the data base, or who may use it for testing or operational purposes. Such instructions include criteria, procedures, and formats for:

a. Submitting data for entry into the data base and identification of a data control organization.
b. Entering data into the data base.

Where these instructions are extensive, reference appropriate sections of other documents.

2.5. Support Software. Describe briefly all support software directly related to the data base. Descriptions should include name, function, major operating characteristics, and machine run instructions for using the support software. Cite the support software documentation by title, number, and appropriate sections.

Examples of support software are:

a. Data base management systems.
b. Storage allocation software.
c. Data base loading software programs.
d. File processing programs.
e. Other generating, modifying, or updating software.

3. LOGICAL CHARACTERISTICS

A data base is a logical arrangement of data. Sets (aggregates), files, records, elements, and items of data may vary in their logical arrangement and relationships. The organization of the content of this section should provide a meaningful presentation of the logical organization of the data base.

Define each unique set (aggregate), file, record, element, or item of data providing information, such as:

a. **Identification.** Name and tag, or label.
b. **Definition.** Standard or unique; purpose in data base; using software; media; form; format and size; update criteria and conditions; security and privacy restrictions, limitations, or conditions (update or access); integrity and validity characteristics; controlling data elements or items; and graphic representation.
c. **Relationships.** Superior and inferior relationships; update and access relationships.

4. PHYSICAL CHARACTERISTICS

4.1. Storage. Specify the storage requirements for the data base and any constraints and conditions.

a. Internal. Describe and illustrate the use of internal storage areas set aside for data including indexing and working areas. Briefly state the equipment constraints and design considerations that affect the use of internal storage.
b. Device. List by device type all peripheral storage required for the data base. Briefly state any contraints imposed on storage requirements by each storage device. State requirements for permanent data storage and temporary data storage, including overlays.
c. Offline. Describe the form, media and storage requirements of all offline data storage.

4.2. Access. Describe the access method and specify the physical relationships of access (index, device, area). Describe all physical access security mechanisms.

4.3. Design Considerations. State the design considerations for the handling of this data base, such as blocking factors. Emphasize those physical relationships important to the efficient utilization of the data base.

See Examples of Content Organization for Section 3 on page 36.

EXAMPLES OF CONTENT ORGANIZATION FOR SECTION 3

Example A: Simple structure in which the data base is composed only of data elements:

Element 1 (Identification, Definition, Relationships)
Element 2 (Identification, Definition, Relationships)
Element N (Identification, Definition, Relationships)

Example B: Simple hierarchial structure in which the data base is composed of files, records, and data elements:

File 1 (Identification, Definition, Relationships)
  Record 1 (Identification, Definition, Relationships)
    Element 1 (Identification, Definition, Relationships)
    Element N (Identification, Definition, Relationships)
  Record N (Identification, Definition, Relationships)
File N (Identification, Definition, Relationships)

Example C: A structure in which a data base is composed of data elements and sets of data with an organization based on multiple or specific relationships between elements and sets:

Element 1 (Identification, Definition, Relationships)
Element N (Identification, Definition, Relationships)

Set 1 (Identification, Definition, Relationships)
Set N (Identification, Definition, Relationships)

Example D: Any of the above structures, but with a substantial number of sets, files, records, elements, or items of data. Outline in graph or chart form the structure, levels, and relationships with each chart element denoting the Identification of the set, etc., portrayed. Supplement the graph or chart with a suitably organized listing of all sets, etc., with the appropriate Definition and Relationships information.

The purpose of the Users Manual is to sufficiently describe the functions performed by the software in non-ADP terminology, such that the user organization can determine its applicability and when and how to use it. It should serve as a reference document for preparation of input data and parameter, and interpretation of results.

Contents

	Page
SECTION 1. GENERAL INFORMATION	2
1.1. Summary	2
1.2. Environment	2
1.3. References	2
SECTION 2. APPLICATION	2
2.1. Description	2
2.2. Operation	2
2.3. Equipment	2
2.4. Structure	2
2.5. Performance	2
2.6. Data Base	2
2.7. Inputs, Processing, and Outputs	2
SECTION 3. PROCEDURES AND REQUIREMENTS	3
3.1. Initiation	3
3.2. Input	3
3.2.1. Input Formats	3
3.2.2. Sample Inputs	3
3.3. Output	4
3.3.1. Output Formats	4
3.3.2. Sample Outputs	4
3.4. Error and Recovery	4
3.5. File Query	4

Users Manual

1. GENERAL INFORMATION

1.1. Summary. Summarize the application and general functions of the software.

1.2. Environment. Identify the user organization and computer center where the software is installed.

1.3. References. List applicable references, such as:

a. Project request (authorization).
b. Previously published documents on the project.
c. Documentation concerning related projects and software.
d. FIPS publications and other reference documents.

2. APPLICATION

2.1. Description. Describe when and how the software is used and the unique support provided to the user organization. The description should include:

a. Purpose of the software.
b. Capabilities and operating improvements provided.
c. Functions performed.

2.2. Operation. Show the operating relationships of the functions performed to the organization that provides input to and receives output from the software. Describe security and privacy considerations. Include general charts and a description of the inputs and outputs shown on the charts.

2.3. Equipment. Describe the equipment on which the software can be run.

2.4. Structure. Show the structure of the software and describe the role of each component in the operation of the software.

2.5. Performance. Describe the performance capabilities of the software including where appropriate:

a. Quantitative information on inputs, outputs, response time, processing times, and error rates.
b. Qualitative information about flexibility and reliability.

2.6. Data Base. Describe all data files in the data base that are referenced, supported, or kept current by the software. The description should include the purpose for which each data file is maintained.

2.7. Inputs, Processing, and Outputs. Describe the inputs, the flow of data through the processing cycle, and the resultant outputs. Include any applicable relationships among inputs or outputs.

3. PROCEDURES AND REQUIREMENTS

This section should provide information about initiation procedures, and preparation of data and parameter inputs for the software. The scope, quality, and logical arrangement of the information should enable the user to prepare required inputs and should explain in detail the characteristics and meaning of the outputs. It should also describe error, recovery, and file query procedures and requirements.

3.1. Initiation. Describe step-by-step procedures required to initiate processing.

3.2. Input. Define the requirements of preparing input data and parameters. Typical considerations are:

a. Conditions—e.g., personnel transfer, out of stock.
b. Frequency—e.g., periodically, randomly, as a function of an operational situation.
c. Origin—e.g., Personnel Section, Inventory Control.
d. Medium—e.g., keyboard, punched card, magnetic or paper tape.
e. Restrictions—e.g., priority and security handling, limitations on what files may be accessed by this type of transaction.
f. Quality control—e.g., instructions for checking reasonableness of input data, action to be taken when data appears to be in error, documentation of errors.
g. Disposition—e.g., instructions necessary for retention or release of all data files received, other recipients of the inputs.

3.2.1. Input Formats. Provide the layout forms used in the initial preparation program data and parameter inputs. Explain each entry, and reference it to the sample form. Include a description of the grammatical rules and conventions used to prepare input, such as:

a. Length—e.g., characters/line, characters/item.
b. Format—e.g., left justified.
c. Labels—e.g., tags or identifiers.
d. Sequence—e.g., the order and placement of items in the input.
e. Punctuation—e.g., spacing and use of symbols (virgule, asterisk, character combinations, etc.) to denote start and end of input, of lines, of data groups, etc.
f. Combination—e.g., rules forbidding use of groups of particular characters, or combinations of parameters in an input.
g. Vocabulary—e.g., an appendix which lists the allowable character combinations or codes that must be used to identify or compose input items.
h. Omissions and Repeats—e.g., indicate those elements of input that that are optional or may be repeated.
i. Controls—e.g., header or trailer control data.

3.2.2. Sample Inputs. Provide specimens of each complete input form. Include:

a. Control or header—e.g., entries that denote the input class or type, date/time, origin, and instruction codes to the software.
b. Text—e.g., subsections of the input representing data for operational files, request parameters for an information retrieval program.

c. Trailer—e.g., control data denoting the end of input and any additional control data.
d. Omissions—e.g., indicate those classes or types of input that may be omitted or are optional.
e. Repeats—e.g., indicate those positions of the input that may be repeated.

8.3. Output. Describe the requirements relevant to each output. Typical considerations are:

a. Use—e.g., by whom and for what.
b. Frequency—e.g., weekly, periodically, or on demand.
c. Variations—e.g., modifications that are available to the basic output.
d. Destination—e.g., computer area, remote terminal.
e. Medium—e.g., printout, CRT, tape, cards.
f. Quality control—e.g., instructions for identification, reasonableness checks, editing and error correction.
g. Disposition—e.g., instructions necessary for retention or release, distribution, transmission, priority, and security handling.

3.3.1. Output Formats. Provide a layout of each output. Explanations should be keyed to particular parts of the format illustrated. Include:

a. Header—e.g., title, identification, date, number of output parts.
b. Body—e.g., information that appears in the body or text of the output, columnar headings in tabular displays, and record layouts in machine readable ouputs. Note which items may be omitted or repeated.
c. Trailer—e.g., summary totals, trailer labels.

3.3.2. Sample Outputs. Provide a sample of each type of output. For each item on a sample, include:

a. Definition—e.g., the meaning and use of each information variable.
b. Source—e.g., the item extracted from a specific input, from a data base file, or calculated by software.
c. Characteristics—e.g., the presence or absence of the item under certain conditions of the output generation, range of values, unit of measure.

3.4. Error and Recovery. List error codes or conditions generated by the software and corrective action to be taken by the user. Indicate procedures to be followed by the user to ensure that any restart and recovery capability can be used.

3.5. File Query. Prepare this paragraph for software with a file query retrieval capability. Include detailed instructions necessary for initiation, preparation, and processing of a query applicable to the data base. Describe the query capabilities, forms, commands used, and control instructions required.

If the software is queried through a terminal, provide instructions for terminal operators. Describe terminal setup or connect procedures, data or parameter input procedures, and control instructions. Reference related materials describing query capabilities, languages, installation conventions and procedures, program aids, etc.

The purpose of the Operations Manual is to provide computer operations personnel with a description of the software and of the operational environment so that the software can be run.

Contents

	Page
SECTION 1. GENERAL INFORMATION	2
1.1. Summary	2
1.2. Environment	2
1.3. References	2
SECTION 2. OVERVIEW	2
2.1. Software Organization	2
2.2. Program Inventory	2
2.3. File Inventory	2
SECTION 3. DESCRIPTION OF RUNS	2
3.1. Run Inventory	2
3.2. Run Progression	2
3.3. Run Description (Identify)	2
3.3.1. Control Inputs	2
3.3.2. Operating Information	2
3.3.3. Input-Output Files	3
3.3.4. Output Reports	3
3.3.5. Reproduced Output Reports	3
3.3.6. Restart/Recovery Procedures	3
3.4. Run Description (Identify)	3
SECTION 4. NON-ROUTINE PROCEDURES	3
SECTION 5. REMOTE OPERATIONS	3

Operations Manual

1. GENERAL INFORMATION

1.1. Summary. Summarize the general functions of the software.

1.2. Environments. Identify the software sponsor, developer, user organization, and the computer center where the software is to be installed.

1.3. References. List applicable references, such as:

a. Project request (authorization).
b. Previously published documents on the project.
c. Documentation concerning related projects.
d. FIPS publications and other reference documents.

2. OVERVIEW

2.1. Software Organization. Provide a diagram showing the inputs, outputs, data files, and sequence of operations of the software. Runs may be grouped by periods of time cycles, by organizational level where they will be performed, or by other groupings.

2.2. Program Inventory. Identify each program by title, number, and mnemonic reference.

2.3. File Inventory. Identify each permanent file that is referenced, created, or updated by the system. Include the title, mnemonic reference, storage medium, and required storage.

3. DESCRIPTION OF RUNS

3.1. Run Inventory. List the various runs possible and summarize the purpose each run. Show the programs that are executed during each run.

3.2. Run Progression. Describe the manner in which progression advances from one run to another so that the entire run cycle is completed.

3.3. Run Description (Identify). Organize the information on each run into the most useful presentation for the operating center and operations personnel involved.

3.3.1. Control Inputs. List the run stream control statements needed for the run.

3.3.2. Operating Information. Provide information for the operating center personnel and management, such as:

a. Run identification.
b. Operating requirements.
c. Initiation method, such as on request, at predetermined time, etc.
d. Estimated run time and turnaround time.
e. Operator commands and messages.
f. Contacts for problems with the run.

3.3.3. Input-Output Files. Provide information for files created or updated by the run, such as:

a. File name or label.
b. Recording medium.
c. Retention schedule.
d. Disposition of file.

3.3.4. Output Reports. For each output report or type of report, provide information such as:

a. Report identification.
b. Medium.
c. Volume of report.
d. Number of copies.
e. Distribution.

3.3.5. Reproduced Output Reports. For those reports that are computer- generated and then reproduced by other means, provide information such as:

a. Report identification.
b. Reproduction technique.
c. Dimensions of paper or other medium.
d. Binding method.
e. Distribution.

3.3.6. Restart/Recovery Procedures. Describe procedures to restart the run or recover from a failure.

3.4 **Run Description (Identify).** Present information about the subsequent runs in a manner similar to that used in paragraph 3.3.

4. NON-ROUTINE PROCEDURES

Provide any information necessary concerning emergency or non-routine operations, such as:

a. Switchover to a back-up system.
b. Procedures for turnover to maintenance programmers.

5. REMOTE OPERATIONS

Describe the procedures for running the programs through remote terminals.

The purpose of the Program Maintenance Manual is to provide the maintenance programmer with the information necessary to understand the programs, their operating environment, and their maintenance procedures.

Contents

	Page
SECTION 1. GENERAL INFORMATION	2
1.1. Summary	2
1.2. Environment	2
1.3. References	2
SECTION 2. PROGRAM DESCRIPTIONS	2
2.1. Program (Identify) Description	2
2.1.1. Problem and Solution Method	2
2.1.2. Input	2
2.1.3. Processing	2
2.1.4. Output	2
2.1.5. Interfaces	2
2.1.6. Tables	2
2.1.7. Run Description	3
2.2. Program (Identify) Description	3
SECTION 3. OPERATING ENVIRONMENT	3
3.1. Hardware	3
3.2. Support Software	3
3.2.1. Operating System	3
3.2.2. Compiler/Assembler	3
3.2.3. Other Software	3
3.3. Data Base	3
SECTION 4. MAINTENANCE PROCEDURES	3
4.1. Programming Conventions	3
4.2. Verification Procedures	3
4.3. Error Correction Procedures	3
4.4. Special Maintenance Procedures	3
4.5. Listings and Flowcharts	3

Program Maintenance Manual

1. GENERAL INFORMATION

1.1. Summary. Summarize the general nature of the software to be maintained.

1.2. Environment. Identify the project sponsor, developer, user and computer center or network where the software is implemented.

1.3. References. List applicable references, such as:

a. Project request (authorizations).
b. Previously published documents on the project.
c. Documentation concerning related projects.
d. FIPS publications and other reference documents.

2. PROGRAM DESCRIPTIONS

Describe the program and programs in the system/subsystem for the maintenance programmer. If a complex system is being described, provide a general description of that system identifying each program and its functions.

2.1. Program (Identify) Description. Identify the program by title, tag or label, and programming language.

2.1.1. Problem and Solution Method. Describe the problem to be solved or the program function and the solution method used.

2.1.2. Input. Describe the input to the program and provide a layout. Identify the medium used. Include information, such as codes, units of measurement, format, range of values, or reference a data element directory.

2.1.3. Processing. Describe processing features and purposes important to the maintenance programmer, such as:

a. Processing logic.
b. Linkages.
c. Variables and constants.
d. Formulas.
e. Error handling provisions.
f. Restrictions and limitations.
g. Locations, settings, internal switches and flags.
h. Shared storage.

2.1.4. Output. Describe the output of the program and provide a layout. Identify the medium used.

2.1.5. Interfaces. Describe the interfaces with other software, such as data formats, messages, parameters, conversion requirements, interface procedures, and media.

2.1.6. Tables. Identify each table and its items. Describe the location, structure, and purpose of each.

2.1.7. Run Description. Describe or reference the operating procedures to run the program, including loading, operating, terminating, and error handling.

2.2. **Program (Identify) Description.** Describe the second through nth computer program in a manner similar to that used in paragraph 2.1.

3. OPERATING ENVIRONMENT

3.1. **Hardware.** Identify the equipment required for the operation of the system. Describe any unusual features used. Relate the hardware to each program. Include information such as:

a. Processor and size of internal storage.
b. Storage online or offline, media, form, and devices.
c. Input/output devices, online and offline.
d. Data transmission devices.

3.2. **Support Software.** Identify the support software needed for each computer program.

3.2.1. Operating System. Identify and describe the operating system including the version or release number and any unusual features used.

3.2.2. Compiler/Assembler. Identify and describe the compiler or assembler including the version or release number and any special features used.

3.2.3. Other Software. Identify and describe any other software used including data management systems, report generators, etc.

3.3. **Data Base.** Describe or reference documentation on the data base used. Include information such as codes, units of measurement, format, range of values, or reference a data element directory.

4. MAINTENANCE PROCEDURES

4.1. **Programming Conventions.** Identify and describe the programming conventions used.

4.2. **Verification Procedures.** Describe the verification procedures to check the performance of the programs, either general or following modifications. Include a reference to test data and testing procedures.

4.3. **Error Correction Procedures.** Describe all error conditions, their sources, and procedures for their correction.

4.4. **Special Maintenance Procedures.** Describe any special procedures required for the maintenance of the programs. Include information such as periodic purges of the data base, temporary modifications needed for leap years or century changes, etc.

4.5. **Listings and Flowcharts.** Reference, append, or describe the method for obtaining copies of listings of the programs and flowcharts.

The purpose of the Test Plan is to provide a plan for the testing of software; detailed specifications, descriptions, and procedures for all tests; and test data reduction and evaluation criteria.

Contents

	Page
SECTION 1. GENERAL INFORMATION	2
1.1. Summary	2
1.2. Environment and Pretest Background	2
1.3. References	2
SECTION 2. PLAN	2
2.1. Software Description	2
2.2. Milestones	2
2.3. Testing (Identify Location)	2
2.3.1. Schedule	2
2.3.2. Requirements	2
2.3.3. Testing Materials	2
2.3.4. Test Training	2
2.4. Testing (Identify Location)	3
SECTION 3. SPECIFICATIONS AND EVALUATION	3
3.1. Specifications	3
3.1.1. Requirements	3
3.1.2. Software Functions	3
3.1.3. Test/Function Relationships	3
3.1.4. Test Progression	3
3.2. Methods and Constraints	3
3.2.1. Methodology	3
3.2.2. Conditions	3
3.2.3. Extent	3
3.2.4. Data Recording	3
3.2.5. Constraints	3
3.3. Evaluation	3
3.3.1. Criteria	3
3.3.2. Data Reduction	3
SECTION 4. TEST DESCRIPTIONS	3
4.1. Test (Identify)	3
4.1.1. Control	3
4.1.2. Inputs	4
4.1.3. Outputs	4
4.1.4. Procedures	4
4.2. Test (Identify)	4

Test Plan

1. GENERAL INFORMATION

1.1. **Summary.** Summarize the functions of the software and the tests to be performed.

1.2. **Environment and Pretest Background.** Summarize the history of the project. Identify the user organization and computer center where the testing will be performed. Describe any prior testing and note results that may affect this testing.

1.3. **References.** List applicable references, such as:

a. Project request (authorization).
b. Previously published documents on the project.
c. Documentation concerning related projects.
d. FIPS publications and other reference documents.

2. PLAN

2.1. **Software Description.** Provide a chart and briefly describe the inputs, outputs, and functions of the software being tested as a frame of reference for the test descriptions.

2.2. **Milestones.** List the locations, milestones events, and dates for the testing.

2.3. **Testing (Identify Location).** Identify the participating organizations and the location where the software will be tested.

2.3.1. Schedule. Show the detailed schedule of dates and events for the testing at this location. Such events may include familiarization, training, data, as well as the volume and frequency of the input.

2.3.2. Requirements. State the resource requirements, including:

a. Equipment. Show the expected period of use, types, and quantities of the equipment needed.
b. Software. List other software that will be needed to support the testing that is not part of the software to be tested.
c. Personnel. List the numbers and skill types of personnel that are expected to be available during the test from both the user and development groups. Include any special requirements such as multishift operation or key personnel.

2.3.3. Testing Materials. List the materials needed for the test, such as:

a. Documentation.
b. Software to be tested and its medium.
c. Test inputs and sample outputs.
d. Test control software and worksheets.

2.3.4. Test Training. Describe or reference the plan for providing training in the use of the software being tested. Specify the types of training, personnel to be trained, and the training staff.

2.4. **Testing (Identify Location).** Describe the plan for the second and subsequent locations where the software will be tested in a manner similar to paragraph 2.3.

3. SPECIFICATIONS AND EVALUATION

3.1. Specifications.

3.1.1. Requirements. List the functional requirements established by earlier documentation.

3.1.2. Software Functions. List the detailed software functions to be exercised during the overall test.

3.1.3. Test/Function Relationships. List the tests to be performed on the software and relate them to the functions in paragraph 3.1.2.

3.1.4. Test Progression. Describe the manner in which progression is made from one test to another so that the entire test cycle is completed.

3.2. Methods and Constraints.

3.2.1. Methodology. Describe the general method or strategy of the testing.

3.2.2. Conditions. Specify the type of input to be used, such as live or test data, as well as the volume and frequency of the input.

3.2.3. Extent. Indicate the extent of the testing, such as total or partial. Include any rationale for partial testing.

3.2.4. Data Recording. Discuss the method to be used for recording the test results and other information about the testing.

3.2.5. Constraints. Indicate anticipated limitations on the test due to test conditions, such as interfaces, equipment, personnel, data bases.

3.3. Evaluation.

3.3.1. Criteria. Describe the rules to be used to evaluate test results, such as range of data values used, combinations of input types used, maximum number of allowable interrupts or halts.

3.3.2. Data Reduction. Describe the techniques to be used for manipulating the test data into a form suitable for evaluation, such as manual or automated methods, to allow comparison of the results that should be produced to those that are produced.

4. TEST DESCRIPTIONS

4.1. Test (Identify). Describe the test to be performed.

4.1.1. Control. Describe the test control, such as manual, semi-automatic, or automatic insertion of inputs, sequencing of operations, and recording of results.

4.1.2. Inputs. Describe the input data and input commands used during the test.

4.1.3. Outputs. Describe the output data expected as a result of the test and any intermediate messages that may be produced.

4.1.4. Procedures. Specify the step-by-step procedures to accomplish the test. Include test setup, initialization, steps, and termination.

4.2. **Test (Identify).** Describe the second and subsequent tests in a manner similar to that used in paragraph 4.1.

3.10 Test Analysis Report

The purpose of the Test Analysis Report is to document the test analysis results and findings; present the demonstrated capabilities and deficiencies for review; and provide a basis for preparing a statement of software readiness for implementation.

Contents

	Page
SECTION 1. GENERAL INFORMATION	2
1.1. Summary	2
1.2. Environment	2
1.3. References	2
SECTION 2. TEST RESULTS AND FINDINGS	2
2.1. Test (Identify)	2
2.1.1. Dynamic Data Performance	2
2.1.2. Static Data Performance	2
2.N. Test (Identify)	2
SECTION 3. SOFTWARE FUNCTION FINDINGS	2
3.1. Function (Identify)	2
3.1.1. Performance	2
3.1.2. Limits	2
3.N. Function (Identify)	3
SECTION 4. ANALYSIS SUMMARY	3
4.1. Capabilities	3
4.2. Deficiencies	3
4.3. Recommendations and Estimates	3

Test Analysis Report

1. GENERAL INFORMATION

1.1. Summary. Summarize both the general functions of the software tested and the test analysis performed.

1.2. Environment. Identify the software sponsor, developer, user organization, and the computer center where the software is to be installed. Assess the manner in which the test environment may be different from the operational environment and the effects of this difference on the tests.

1.3. References. List applicable references, such as:

a. Project request (authorization).
b. Previously published documents on the project.
c. Documentation concerning related projects.
d. FIPS publications and other reference documents.

2. TEST RESULTS AND FINDINGS

Identify and present the results and findings of each test separately in paragraphs 2.1 through 2.N.

2.1. Test (Identify).

2.1.1. Dynamic Data Performance. Compare the dynamic data input and output results, including the output of internally generated data, of this test with the dynamic data input and output requirements. State the findings.

2.1.2. Static Data Performance. Compare the static data input and output results, including the output of internally generated data, of this test with the static data input and output requirements. State the findings.

2.N. Test (Identify). Present the results and findings of the second and succeeding tests in a manner similar to that of paragraph 2.1.

3. SOFTWARE FUNCTION FINDINGS

Identify and describe the findings on each function separately in paragraphs 3.1 through 3.N.

3.1. Function (Identify).

3.1.1. Performance. Describe briefly the function. Describe the software capabilities that were designed to satisfy this function. State the findings as to the demonstrated capabilities from one or more tests.

3.1.2. Limits. Describe the range of data values tested, including both dynamic and static data. Identify the deficiencies, limitations, and constraints detected in the software during the testing with respect to this function.

3.N. **Function (Identify).** Present the findings on the second and succeeding functions in a manner similar to that of paragraph 3.1.

4. ANALYSIS SUMMARY

4.1. **Capabilities.** Describe the capabilities of the software as demonstrated by the tests. Where tests were to demonstrate fulfillment of one or more specific performance requirements, prepare findings showing the comparison of the results with these requirements. Assess the effects any differences in the test envirment as compared to the operational environment may have had on this test demonstration of capabilities.

4.2. **Deficiencies.** Describe the deficiencies of the software as demonstrated by the tests. Describe the impact of each deficiency on the performance of the software. Describe the cumulative or overall impact on performance of all detected deficiencies.

4.3. **Recommendations and Estimates.** For each deficiency provide any estimates of time and effort required for its correction and any recommendations as to:

a. The urgency of each correction.
b. Parties responsible for corrections.
c. How the corrections should be made.

State the readiness for implementation of the software.

U.S. GOVERNMENT PRINTING OFFICE : 1976 O—200-393